12-12-2019

# Low-Rank Multivariate General Linear Model with Relationship between Brain Regions and Time Points

Ge Song
gs4799@rit.edu


# Low-Rank Multivariate General Linear Model with Relationship between Brain Regions and Time Points

**By**

**Ge Song**

A thesis submitted in partial fulfillment of the requirement for the Degree of Master of Science in Applied Statistics

Department of Applied Statistics

College of Science

**Rochester Institute of Technology**

Rochester, NY

Dec 12, 2019

## Committee Approval:

---

Dr. Peter Bajorski Date

Thesis Advisor/Committee Member

---

Dr. Minh Pham Date

Thesis Advisor /Committee Member

---

Dr. Robert Parody Date

Committee Member

# Abstract

The human brain is hard to study and analysis, not because of the complexity of the brain structure, such as neurons and neurons connections, but also because of the complexity of the brain activities. Since in different scales, for example, different time series, different physical senses, the measurements of the human brain activities can be varied.  Tring to measure the brain regions relationships in person, the Functional Magnetic Resonance Imaging (fMRI) is one of the methods. The focus of this paper is on analyzing human brain regions relationships in different time domains and different scans of fMRI by using low-rank multivariate general linear model (LRMGLM). The function of the model is to penalize optimization and characterize variation across different regions and stimulus in hemodynamic response functions (HRFs). After analyzing the fMRI data with LRMGLM model, we also analyzed data by methods of Cross Validation and Principal Components Analysis (PCA).

# Acknowledgement

First, thanks to Professor Peter Bajorski and Professor Minh Pham being my thesis advisors. Thank them for always giving me guidance and advising. During my thesis year, they were always patient to me and supported my work; and they also told me how to do professionally in a research work and help me without hesitation when I struggled in my thesis. I cannot imagine what I would be without their help. Sincerely thank them to be my advisors and changed me.

Especially thank Professor Robert Parody to be my committee member. I really enjoyed when I in his classes and learned a lot. In my thesis work, I used a lot of knowledge taught by him and he always welcome me to ask questions and gave me advice.

Thank Shawna Hayes who always kept me in right timeline. Shawna help me a lot to finish my thesis and even degree. She always tried her best to answer my questions about the thesis and tried to keep me in the most professional way to finish my thesis.

Then, I want to thank my parents who always support me and let me fully focus on my study without other worries. They actually did not understand what I did in my thesis, but they told me that if I have problems, try to think the problems in a good way and then I will in a right way to solve the problems. My family is the most important for me and thank my family help me to chase me dream.

Also, thank my best friend, roommate, classmate and partner, Yidan Yang. We did the thesis together at the beginning, and after we work separately, she always help me and gave me coding advises. We are usually kidding to each other that I will be your side when I need help. And I hope she is doing well in her PhD years.

# Table of Content

# Introduction

Although many scientists in various areas investigate how human brains active and the mapping of human brains over a century, it still has remained challenging since the complexity of human brain and no one can easily interpret that how the brain active and what does each part of brain work for. Based on the biological properties, like molecular, cellular, cortex of the brain, the areas or the regions of brain can be considered relatively functional and connected. By using the technology tools, we may find some connectives between brain regions and its function.

Functional magnetic resonance imaging (fMRI) is used to measure the blood flow occur during the brain activity and mapping brain activity. The active of neurons in our brain change in different activities, and blood flow to brain areas also change in neural activities. The fMRI scans can provide the anatomical information of brain activities about the patterns and regions by visual version. The scan of fMRI can tell us the shape, size and integrity in gray and white structures.

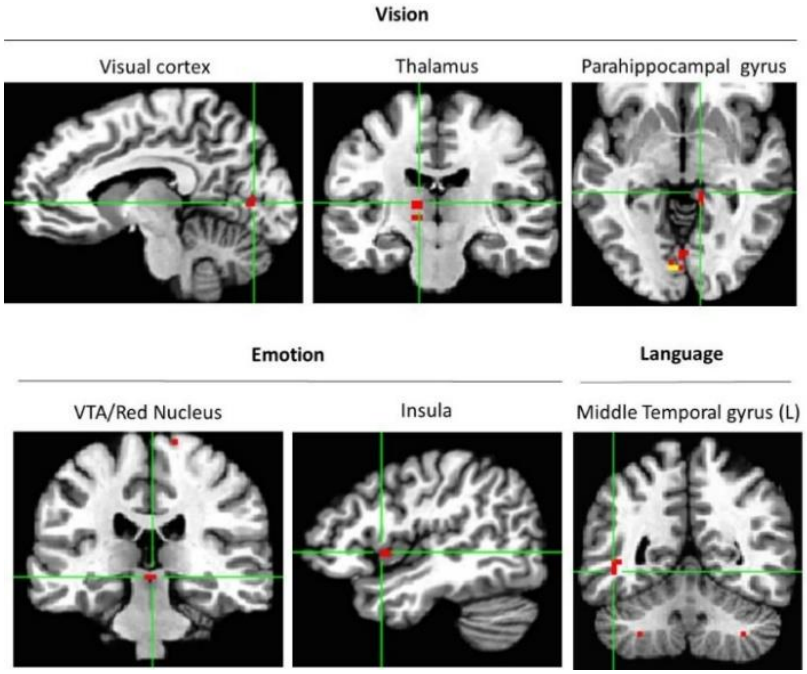
In this thesis, we analyzed the fMRI data from 820 people and their scans. The data is four dimensional – [1200, 4, 116, 820], and for each dimension, they refer to time series (1200), scans (4), regions (116) and subjects (820). We did the cross validation for 820 subjects with their first scan based on the Low-Rank Multivariate General Linear Model and got the object values of the parameters in the model, then selected top 2 principal components after the Principal Component Analysis (PCA). We also analyzed the relationships of subjects' regions: we focused on first several people with their 116 regions after ordered in all 4 scans.

# Related Works

## fMRI

fMRI (Functional Magnetic Resonance Imaging) is a technique of measuring the brain activity. It reflects the flow of blood in human brain instead of recording the actual activity of neurons, so that gives the activity patterns of neurons. fMRI is a type of MRI (Magnetic Resonance Imaging), which fMRI is based on the same technology and one step farther of MRI. The difference between fMRI and MRI is that MRI creates images of organs and tissues by using magnetic field and radio waves in the test and it is a tool to detect diseases of brain; but fMRI is more detail: not only can it detect diseases, it also determines the regions of our brain, such as which parts are for talking, thinking, moving and so on by examining the changing of the blood flow in the parts of our brain.

fMRI has some advantages than MRI. First, it is no risks typically since it does not use radiation, so that it is non-invasively. Second, the spatial resolution is good for experimental research by fMRI scan gives detail by millimeter. fMRI can also provide the local of how brain activity: different blood flows are due to varies of neuronal activity which the changes can reflect the brain activity in specific brain regions– also known as blood oxygenation level dependent (BOLD) imaging.



fMRI mapping now is popular when scientists want to identify and delineate the regions of the human brain by the different levels of stimuli and tasks during the brain activities. With the purpose of detecting what relationships and functions between brain regions more detail, different time points, different subjects or different fMRI scans are needed for analyzing.

## Low-Rank Multivariate General Linear Model

The Low-Rank Multivariate General Linear Model (LRMGLM) is a new model for stimuli-evoked fMRI data, and the model is used to assess response of brain with different stimuli and distinguish regions of human brain with different responses by using different subjects (people) and stimulus-evoked fMRI data with their fMRI scans. It is also used to describe variation in hemodynamic response functions (HRFs) through different regions of brain and different kinds of stimulus.

Notations of Parameters:

| Parameter | Description |
|---|---|
| $Y^i$ | A T × J matrix for J voxels of the ith subject for the fMRI data. |
| $X^i$ | A T × L design matrix of ith subject. |
| D | A T × r matrix with $t$th row. |
| $d^i$ | A T × J matrix of drift coefficients with jth column |
| $U^i$ | A L × P temporal matrix with elements $U^i_{lp}$, $l = 1,\dots,L \text{ and } p = 1,\dots,P$ for ith subject with same HRF shapes associated with J voxels. |
| $\bar{U}$ | A L × P population average temporal parameter matrix based on $U^i$. |
| $V^i$ | A P × J spatial matrix with elements $V^i_{pj}$, $p = 1,\dots,P \text{ and } j = 1,\dots,J$ for ith subject related with J voxels to describe voxels specific responses. |
| $\bar{V}$ | A P × J population average spatial parameter matrix based on $V^i$. |
| $E^i$ | A T × J matrix with jth column for the error term. |
| P | A given positive constant (used P = 2 and P = 4 in the practice). |
| R | A L × L matrix with $R_{l1,l2} = \int_0^m b^{(2)}_{l1}(t) b^{(2)}_{l2}(t)dt,\ l_1, l_2 = 1,\dots,L.$ |

The model is a bilinear regression model of $Y^i$ with $X^i$,

$$Y^i = Dd_i + \sum_{k=1}^{K} X_k^i U_k^i V_k^i + E^i.$$

It is a multivariate GLM (MGLM) with low-rank representation which is our Low-Rank Multivariate General Linear Model (LRMGLM). In this model, the temporal and spatial are divided into two parts. $Y^i$ is the T × J matrix with (t, j) elements for J voxels of the ith subject; D is a T × r matrix with $t$th row; $d_i$ is the matrix of T × J of drift coefficients with jth column; $X^i$ is a design matrix in T × L form of ith subject; and Matrix $U^i$ are L × P matrices with elements $U_{lp}^i$, $l = 1, \ldots, L \; and \; p = 1, \ldots, P$, and means principal function across J voxels to describe common shapes for the responses; matrix $V^i$ are P × J matrices with elements $V_{pj}^i$, $p = 1, \ldots, P \; and \; j = 1, \ldots, J$ related with J voxels to describe voxels specific responses; $E^i$ be a T × J matrix with jth column. P is a known positive constant and in practice, we selected P = 2 and P = 4 to define main differences between subjects and voxels.

For the optimization procedure, given by the fixed V, the minimizer for U and d is:

$$\min_{U,\bar{d}} \sum_{i=1}^{n} \left\| Y^i - Dd^i - \sum X^i \bar{U} \bar{V} \right\|_F^2 + \lambda \cdot \sum P(\bar{U}),$$

Where penalty $P(\bar{U})$ is $P(\bar{U}) = \sum_{p=1}^{P} \int_0^m (\sum_{l=1}^{L} \bar{U}_{lp} \cdot b_l^{(2)}(t))^2 dt = \bar{U}^T R \bar{U}$; and R is a L × L matrix with $R_{l1,l2} = \int_0^m b_{l1}^{(2)}(t) b_{l2}^{(2)}(t) dt, \; l_1, l_2 = 1, \ldots, L.$

And this thesis is analyzed from this step and started with the parameter $\lambda$.

# Statistical Analysis

## Part 1. Preliminary Analysis

The fMRI scan data we used are from the Human Connectome Project. The purpose of the Project is to detect the connective and function of human brain and based on the mapping of the human brain scans to find the relationship of brain regions.

The full fMRI data consists four dimensions and displays in format of array named “Scans.arr”. And each dimension is defined as following:

```
## Dimensions of data

dim(Scans.arr)
## 1200 116 4 820
```

- Dimension 1: 1200 time points for brain activity
- Dimension 2: 116 regions of brain
- Dimension 3: 4 scans of each subject
- Dimension 4: 820 subjects (people)

The following output is the example of first subject of first 4 time points and first 5 regions in all four scans:

```
## Example of 1st subject with the first 4 time poins
## and first 5 regions in all 4 scans
Scans.arr[ 1:4, 1:5, 1:4, 1]
## , , 1
##              [,1]         [,2]         [,3]        [,4]        [,5]
## [1,]  0.1236054431 -0.020382756  0.106195805  0.14163159  0.08466914
## [2,]  0.0800266609  0.010988130  0.007470173  0.03224519 -0.03413749
## [3,]  0.0144267235  0.015400782 -0.079054707 -0.01136957 -0.08003826
## [4,] -0.0001230086  0.002695427 -0.076813373 -0.08833929 -0.03646278

## , , 2
```

```
##            [,1]        [,2]        [,3]       [,4]         [,5]
## [1,] -0.01173671 -0.05686451 0.078815854 0.12241566 -0.009226118
## [2,] -0.07736850 -0.15765601 0.006740389 0.01120843 -0.015375254
## [3,] -0.17109888 -0.20602204 0.041703167 0.01696427 -0.082755036
## [4,] -0.06120633 -0.07538534 0.031231134 0.05435275  0.014704243

## , , 3
##            [,1]      [,2]        [,3]        [,4]         [,5]
## [1,] 0.25676750 0.2797734  0.12961559  0.09123417  0.056552683
## [2,] 0.09558847 0.0911071 -0.06514057 -0.09637130  0.037163975
## [3,] 0.06884158 0.1106374 -0.07470015 -0.07259700  0.005065498
## [4,] 0.09639441 0.1045436 -0.09675553 -0.08255689 -0.030797556

##, , 4
##             [,1]        [,2]        [,3]        [,4]        [,5]
## [1,] -0.018146964 -0.08010770  0.16171339  0.17892989  0.06149853
## [2,] -0.008746116 -0.02117317  0.04397868  0.06688928  0.08870200
## [3,] -0.028190652 -0.02518838 -0.05990051 -0.03432816 -0.10180122
## [4,]  0.029913526  0.03519832 -0.06644190 -0.08965499 -0.08143525
```

And for the following output, it shows the first three subjects each with their first 4 time points and 5 regions in their first scan and ID means each subject's ID number according to the dataset:

```
## Example of first 3 subjects with the first 4 time points
## and first 5 regions in their first scan
Scans.arr[ 1:4, 1:5, 1, 1:3]
## , , ID = 100206
##                 [,1]         [,2]         [,3]        [,4]        [,5]
##   [1,]  0.1236054431 -0.020382756  0.106195805  0.14163159  0.08466914
##   [2,]  0.0800266609  0.010988130  0.007470173  0.03224519 -0.03413749
```

```
##   [3,]  0.0144267235  0.015400782 -0.079054707 -0.01136957 -0.08003826
##   [4,] -0.0001230086  0.002695427 -0.076813373 -0.08833929 -0.03646278

##, , ID = 100307
##               [,1]        [,2]        [,3]        [,4]         [,5]
##   [1,]  0.15101818  0.23247057  0.13967349  0.11074147  0.052400762
##   [2,]  0.08053443  0.01286557  0.11152290  0.06012833  0.037082447
##   [3,] -0.03390613 -0.04879962  0.02367411 -0.04344801 -0.002028682
##   [4,] -0.07632881 -0.05166325 -0.04149430 -0.03555954  0.035835974

##, , ID = 100408
##               [,1]         [,2]        [,3]         [,4]        [,5]
  [1,]  0.10339242  0.166352205  0.06035419  0.147160138  0.05142478
  [2,]  0.03774514 -0.033658780 -0.04003646 -0.002391871 -0.05275825
  [3,]  0.04963075  0.001379108 -0.04984266  0.086536587  0.01633124
  [4,] -0.05103915 -0.081245765 -0.02526441 -0.034715837 -0.02894541
```

In the dataset, every subject has its own ID number and it is repeated four times in the array which is response to each scan of 4 scans; the scan is showed by matrix form – rows are the time points and columns are the regions. Also, each subject recorded their own information. Following is the part of the subjects' information:

| Subject | Release | Acquisitio | Gender | Age | 3T_Full_M | T1_Count | T2_Count | 3T_RS-fMI | 3T_RS-fMI | 3T_Full_Ta |
|---|---|---|---|---|---|---|---|---|---|---|
| 100004 | S900 | Q06 | M | 22-25 | FALSE | 0 | 0 | 0 | 0 | FALSE |
| 100206 | S900 | Q11 | M | 26-30 | TRUE | 1 | 1 | 4 | 100 | TRUE |
| 100307 | Q1 | Q01 | F | 26-30 | TRUE | 1 | 1 | 4 | 100 | TRUE |
| 100408 | Q3 | Q03 | M | 31-35 | TRUE | 1 | 1 | 4 | 100 | TRUE |
| 100610 | S900 | Q08 | M | 26-30 | TRUE | 2 | 1 | 4 | 100 | TRUE |
| 101006 | S500 | Q06 | F | 31-35 | TRUE | 2 | 2 | 4 | 100 | TRUE |
| 101107 | S500 | Q06 | M | 22-25 | TRUE | 2 | 2 | 4 | 100 | TRUE |

Since the subjects' information data saved in "csv" file, for the purpose of using information easily and in the same order with the 4-dimension fMRI dataset, we used the dataframe format in R:

```
## Each subject's information
```

```
load("Brain.subjects.df")

list(Brain.subjects.df)

[[1]]

  Subject Release Acquisition Gender   Age X3T_Full_MR_Compl T1_Count

2  100206    S900         Q11      M 26-30              true        1

  T2_Count X3T_RS.fMRI_Count X3T_RS.fMRI_PctCompl X3T_Full_Task_fMRI

2        1                 4                  100               true

  X3T_tMRI_PctCompl fMRI_WM_PctCompl fMRI_Gamb_PctCompl fMRI_Mot_PctCompl

2               100              100                100               100

  fMRI_Lang_PctCompl fMRI_Soc_PctCompl fMRI_Rel_PctCompl fMRI_Emo_PctCompl

2                100               100               100               100

  X3T_dMRI_Compl X3T_dMRI_PctCompl dMRI_3T_ReconVrs fMRI_3T_ReconVrs

2           true               100             r227             r227

  MEG_AnyData MEG_FullProt_Compl MEG_HeadModel_Avail MEG_CortRibn_Avail

2       false              false               false              false

  MEG_Anatomy_Avail MEG_Anatomy_Compl MEG_Noise_Avail MEG_Noise_Compl

2             false             false           false           false
```

In this thesis, the goal is to find the relationship between regions of human brain by finding the optimal values based on the LRMGLM model for each person's first scan and then analyze for regions with all 4 scans.

## Part 2. Find Optimal Values for model

For the optimization procedure, we already have the minimizer function for U, V and d:

$$\min_{U,\vec{d}} \sum_{i=1}^{n} \left\| Y^i - D d^i - \sum X^i \bar{U} \bar{V} \right\|_F^2 + \lambda \cdot \sum P(\bar{U}),$$

$$=$$

$$\min_{U,\bar{d}} \sum_{i=1}^{n} \left\| Y^i - Dd^i - \sum X^i \bar{U} \bar{V} \right\|_F^2 + \lambda \cdot \sum \bar{U}^T R \bar{U} .$$

So, we need to find the optimal values for model by estimating parameters and try to estimate the model and define regions by minimize the error term. In this case, we can minimize the penalized sum of squared errors (PSSE) by given V, find the minimizer d and U of PSSE:

$$PSSE(\bar{d}, U|V) = SSE(\Theta) + \lambda \cdot \sum \bar{U}^T R \bar{U},$$

where $SSE(\Theta) = \frac{1}{n} \sum_{i=1}^{n} \left\| Y^i - Dd^i - \sum X^i \bar{U} \bar{V} \right\|_F^2, \Theta = \{d^i, \bar{U}, \bar{V}, i = 1, \ldots, n\}$.

And value of V can also be found by given d and U: when fixed d and U, let $W = X \cdot U$, then the function becomes to

$$\min_{\bar{d},V} \sum_{i=1}^{n} \left\| Y^i - Dd^i - \sum W \bar{V} \right\|_F^2 .$$

$\lambda$ is the penalty parameter, so we used the cross validation to choose the best parameter.

**Step 1 - Cross Validation**

Since the full dataset is 3.5GB which is really large, we decided to use the 820 subjects' first scan to do the cross validation first, then the dimension changed to [1200, 116, 820] which is a three dimensions data. Based on the previous algorithm from MATLAB code, we used the same definition for D, B and R from basis dataset and same method in the fucntion: d is random array with dimension [7, 116, 820]; u is random matrix with dimension [33, p = 2] and v is random array with dimension [p = 2, 116, 820].

As the function showed above, when V fixed, d and U can be found by the PSSE function. And since $\lambda$ is a penalty parameter of the function, so first used cross validation to find the $\lambda$.

In practice, the data split into training data and testing data, and we can predict the testing data based on the result of the training data; we choose the range of $\lambda$ between -1 and 15 and P = 4 to test.

```
### Read data from basis from MATLAB
```

```
basis <- readMat("basis.mat")

### Load for 820 subjects' first scan

load("Y_oneScan")

Y <- Y_oneScan

D <- basis$D

B <- basis$B

R <- basis$R

P <- matrix(data = 4, nrow = 1)

lambda1 <- matrix(data = -1:15, nrow = 1)

lambda1 <- exp(lambda1)

error <- matrix(data = 0, nrow = length(P), ncol = length(lambda1))


for (fold in 1:5) {

  # Split data into training and testing

  index = sample(rep(1:5, 820/5))

  train = which(index!=fold)

  test =  which(index==fold)

  Y_train = Y[,,train]

  Y_test = Y[,,test]

  # Choose the number of principal curves

  n = dim(Y_train)[3]

  T = dim(Y_train[,,1])[1]

  J = dim(Y_train[,,1])[2]

  T = dim(B)[1]

  L = dim(B)[2]

  r = dim(D)[2]

  # Initilize starting solution

  # d is random array with dimension (7,116,820)

  # u is random matrix with dimension (33,p) where p=2 in this case

  # v is random array with dimension (p,116,820) where p=2
```

```
  # here n = 820
  # here J = dim(Y_train[,,1])[2]=116
  d_cur = array(data = runif(r*J*n), dim = c(r,J,n))
  for (p in 1:length(P)) {
    U_cur = matrix(data = runif(L*P[p]), nrow = L, ncol = P[p])
    V_cur = array(data = P[p]*J*n, dim = c(P[p],J,n))
    for (i in 1:length(lambda1)){
      lsq = lr.func.stp12(Y_train,D,d_cur,B,U_cur,V_cur,lambda=lambda1[i],R)
      d_cur = lsq$d;U_cur = lsq$u;V_cur = lsq$v
      pre = predict_curve(Y_test,D,B,U_cur)
      d_test=pre$d_test;V_test=pre$V_test;temp=pre$error
      error[p,i] = error[p,i]+temp
    }
  }
}
write.csv(error,file='test_error.csv',row.names = F)
```

The following output shows the result of cross validation and it shows the 17 results for the error term based on each $\lambda$.

| V1 | V2 | V3 | V4 | V5 | V6 | V7 | V8 | V9 | V10 | V11 | V12 | V13 | V14 | V15 | V16 | V17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8620.632 | 8582.057 | 8569.285 | 8567.287 | 8566.819 | 8566.637 | 8566.467 | 8566.36 | 8566.305 | 8566.247 | 8566.245 | 8566.244 | 8566.243 | 8566.244 | 8566.246 | 8566.25 | 8566.309 |

The result of error terms shows a u-shape curve since it is the result of the cross validation. The 13rd result has the smallest error which is when $\lambda = 11$, so took $\lambda = 11$ be the best penalty parameter value in this case.

**Step 2 – Get Object Value**

After the process of choosing $\lambda$ the penalty parameter, the object values of d and U can be found by fixed V and re-run the function

```
## Object Value
obj_val <- function(Y,D,d, B,u,v, lambda, R){
  n = dim(Y)[3]
  val=0;
  for (i in 1:n) {
    val = val + norm(Y[,,i]-D%*%d[,,i]-B%*%u%*%v[,,i],'F')^2
  }
  rob.val = val/n +lambda*sum(diag(t(u)%*%(R%*%u)))
  return(rob.val)
}
op.val <- lr.func.stp12(Y, D, d, B, u, v, lambda = 11, R)
```

The result for the object value of d (example) is

```
## [[1]]
## , , 1
##             [,1]        [,2]        [,3]        [,4]        [,5]
##  [1,]  0.04157544  0.05154041  0.05172297  0.06516508  0.02108517
##  [2,]  0.15862006  0.18315269  0.11588720 -0.06260412 -0.02155352
##  [3,] -0.48475290 -0.65411405 -0.21641134 -0.29636052 -0.11347752
##  [4,]  0.13717319  0.13097238  0.14291411  0.28483455  0.20123680
##  [5,] -0.16736904 -0.13521252 -0.17524334 -0.19576418 -0.24850271
##  [6,] -0.24375751 -0.31284999  0.15557038  0.19128071  0.08928018
##  [7,] -0.18827852 -0.18880764 -0.32750175 -0.33976383 -0.13841079
##              [,6]        [,7]        [,8]          [,9]        [,10]
##  [1,]  0.006300095  0.05946413  0.06644774  0.0367819883  0.011124432
##  [2,]  0.064726182  0.07492595  0.02307757  0.0005665903  0.030041176
##  [3,] -0.118847448 -0.49473647 -0.60142438 -0.4701889634 -0.107655533
##  [4,]  0.016340597  0.27725753  0.39516662  0.2556720025  0.118435509
##  [5,] -0.036110825 -0.34555717 -0.36694505 -0.5066059663 -0.102493347
```

```
##  [6,]  0.070036956  0.04735844  0.12136068 -0.2037391170  0.002904721
##  [7,]  0.021178024 -0.21148159 -0.11531407  0.2535940280  0.178478329
```

The result for the object value of U (example) is

```
## [[1]]
##               [,1]         [,2]
##  [1,] -0.049544193 -0.040528913
##  [2,]  0.021763081  0.006114538
##  [3,]  0.033506046  0.029820103
##  [4,]  0.012275007  0.002094327
##  [5,] -0.031712229 -0.027517157
##  [6,]  0.010616535  0.049424534
##  [7,]  0.041825234 -0.019207856
##  [8,] -0.054240184 -0.021340179
##  [9,]  0.081497655  0.085978878
## [10,] -0.084396198 -0.144311471
## [11,]  0.135594281  0.186072275
## [12,] -0.130227333 -0.151191262
```

Also, we got the values of U and d, then V can also be found by the function:

$$\min_{\vec{d},\bar{V}} \sum_{i=1}^{n} \left\| Y^i - Dd^i - \sum W\bar{V} \right\|_F^2 .$$

The output for V (example) is

```
## [[1]]
## , , 1
##             [,1]       [,2]        [,3]         [,4]        [,5]
## [1,] 0.002551155 0.04748689 -0.11873932 -0.112224036 -0.05676742
## [2,] 0.124341705 0.10874817  0.03636178 -0.007361959  0.10433081
##             [,6]        [,7]        [,8]       [,9]        [,10]
```

```
## [1,] 0.004870743 -0.05228147 -0.01775847 0.04444704 -0.007683727
## [2,] 0.028941801  0.12837545  0.11107678 0.11812994  0.004260577
##           [,11]      [,12]       [,13]       [,14]      [,15]      [,16]
## [1,] 0.09008251 0.06169269 -0.05123165 -0.08083737 -0.2024194 -0.1338652
## [2,] 0.23245898 0.28773865  0.22655447  0.25636684  0.1494012  0.1119079
##           [,17]     [,18]      [,19]       [,20]       [,21]       [,22]
## [1,] 0.06410188 0.1508686 0.04194186  0.01582954 -0.04540883 -0.02610043
## [2,] 0.11230788 0.1381043 0.06258525 -0.02006746 -0.15081725  0.02438789
```

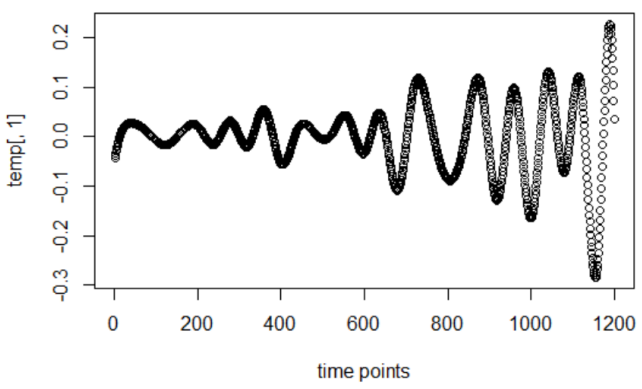
**Step 3 – Check Values with Function**

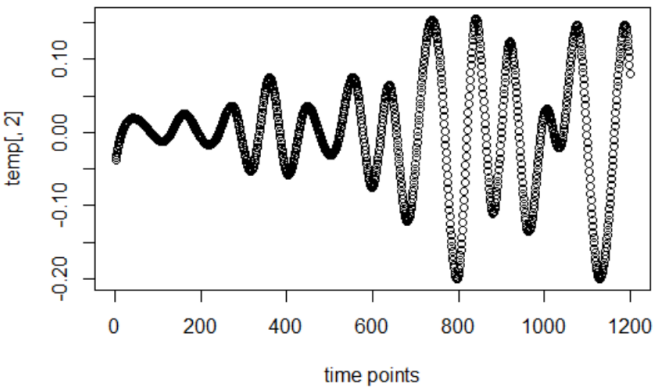
Since we defined $W = X \cdot U$, it is an inverting matrix, so we can check for W to see if there is any information to analysis. (basis$B in the code equal to X in the function, W equals to temp)

```
temp = basis$B %*% op.val$u
temp
##                  [,1]          [,2]
##    [1,] -4.431792e-02 -3.709551e-02
##    [2,] -3.933150e-02 -3.379030e-02
##    [3,] -3.457941e-02 -3.061100e-02
##    [4,] -3.005612e-02 -2.755535e-02
##    [5,] -2.575611e-02 -2.462108e-02
##    [6,] -2.167384e-02 -2.180591e-02
##    [7,] -1.780379e-02 -1.910756e-02
##    [8,] -1.414043e-02 -1.652378e-02
##    [9,] -1.067823e-02 -1.405227e-02
##   [10,] -7.411674e-03 -1.169077e-02
```

The dimension of W (temp) is [1200, 2] which 1200 response to 1200 time points and 2 is the principal curves of brain which is defined previously.

The plots for the two principal curves with 1200 time points are following:



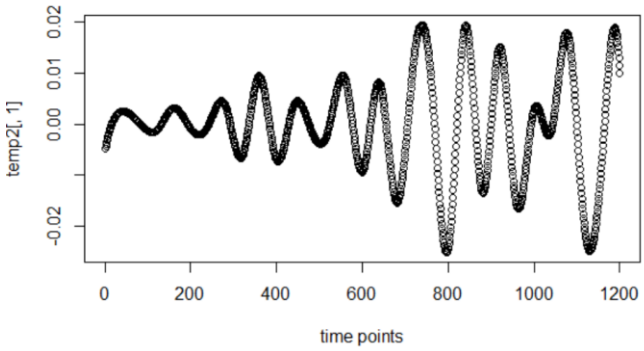



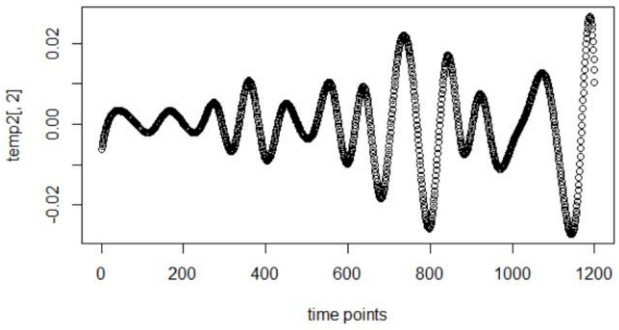

Also, we want to find the term $\sum X^i \bar{U}\bar{V} = \sum W\bar{V}$ for the temporal and spatial parameter matrices. Let temp2 = basis$B %*% op.val$u %*% op.val$v[,,1] to find the relationship between time points and regions.

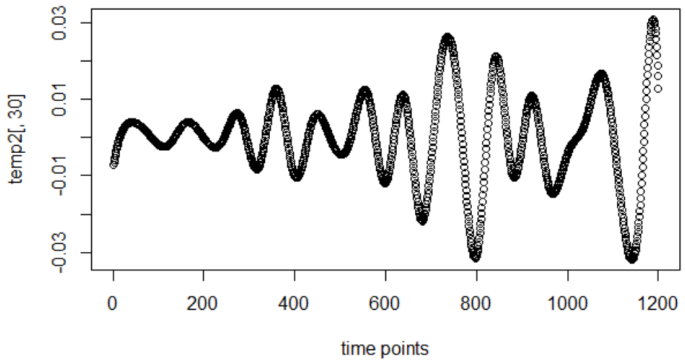
```
temp2 = basis$B %*% op.val$u %*% op.val$v[,,1]

temp2

##                   [,1]          [,2]          [,3]          [,4]
##    [1,] -4.725581e-03 -6.138589e-03  3.913421e-03  5.246631e-03
##    [2,] -4.301884e-03 -5.542363e-03  3.441520e-03  4.662702e-03
##    [3,] -3.894442e-03 -4.970959e-03  2.992865e-03  4.105998e-03
##    [4,] -3.502958e-03 -4.423866e-03  2.566882e-03  3.575881e-03
##    [5,] -3.127135e-03 -3.900575e-03  2.162996e-03  3.071714e-03
##    [6,] -2.766677e-03 -3.400576e-03  1.780635e-03  2.592860e-03
##    [7,] -2.421287e-03 -2.923359e-03  1.419225e-03  2.138682e-03
##    [8,] -2.090669e-03 -2.468416e-03  1.078191e-03  1.708544e-03
##                   [,5]          [,6]          [,7]          [,8]
##    [1,] -1.354391e-03 -1.289472e-03 -2.445147e-03 -3.333431e-03
##    [2,] -1.292621e-03 -1.169526e-03 -2.281536e-03 -3.054850e-03
##    [3,] -1.230687e-03 -1.054365e-03 -2.121839e-03 -2.786094e-03
##    [4,] -1.168664e-03 -9.438972e-04 -1.966053e-03 -2.527009e-03
##    [5,] -1.106630e-03 -8.380298e-04 -1.814175e-03 -2.277441e-03
##    [6,] -1.044660e-03 -7.366700e-04 -1.666203e-03 -2.037236e-03
##    [7,] -9.828326e-04 -6.397250e-04 -1.522134e-03 -1.806239e-03
```

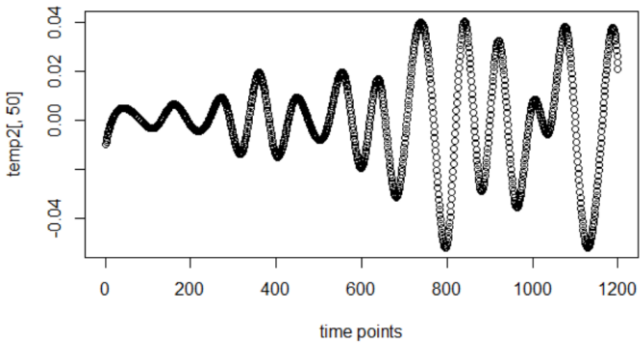
```
##     [8,] -9.212232e-04 -5.471023e-04 -1.381965e-03 -1.584295e-03
```

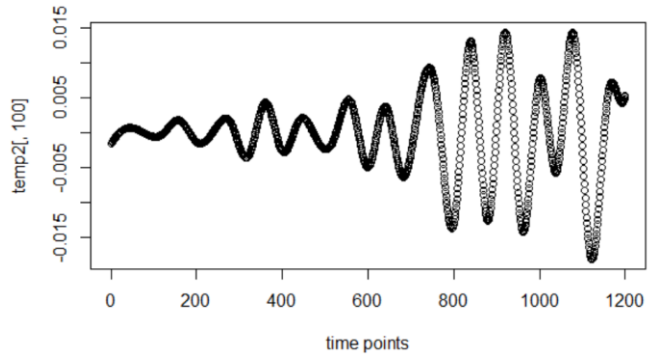
The dimension of temp2 is [1200, 116] which means time points (1200) and regions (116) and the plots (examples) are showed

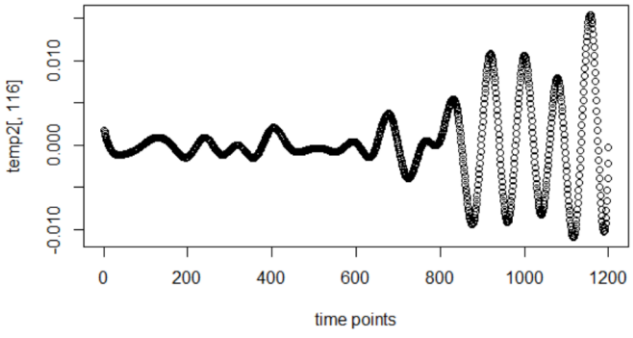



From above plots, we can find that in each plot, the line is more fluctuate at the end than the beginning, especially from time point 800 to time point 1200. By trying to analyze the relationship between time points and regions, we decided to select last 120 time points which is from time point 1081 to time point 1200 to take a look.

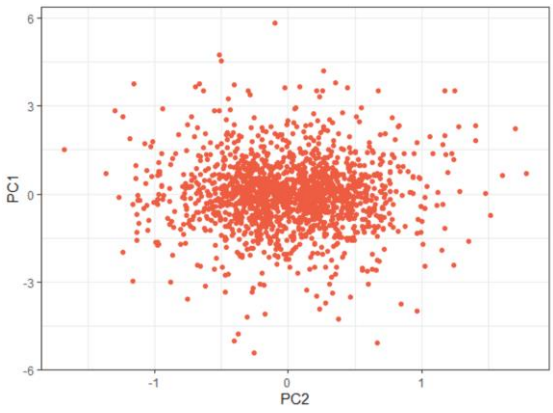
## Part 3. Principal Component Analysis

Before analyzing all 4 scans for each person, Principal Component Analysis is also important in our research. We did the PCA to see clusters of each region for subjects.

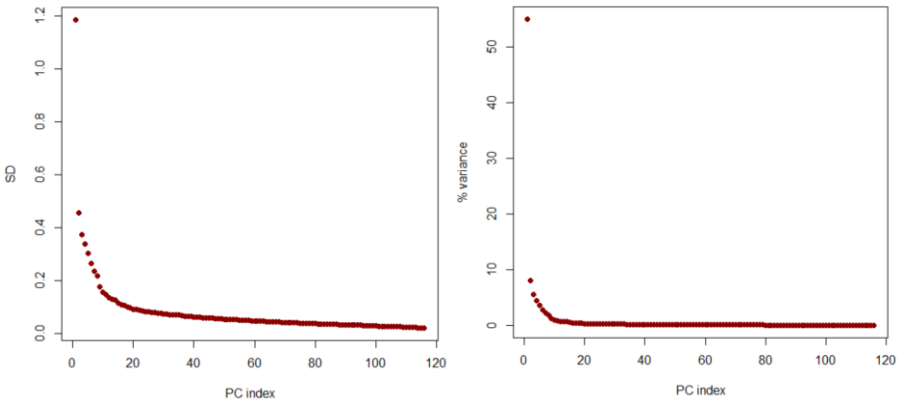
First, look at the importance of components for all principal components (example):

```
PC <- prcomp(~ ., center = T, data = op.vpc)
summary(PC)
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6
## Standard deviation     1.1842 0.45463 0.37417 0.33864 0.30344 0.26489
## Proportion of Variance 0.5499 0.08106 0.05491 0.04497 0.03611 0.02752
## Cumulative Proportion  0.5499 0.63099 0.68590 0.73087 0.76698 0.79450
##                           PC7     PC8     PC9    PC10    PC11    PC12
## Standard deviation     0.23355 0.21696 0.17531 0.15669 0.14536 0.13589
## Proportion of Variance 0.02139 0.01846 0.01205 0.00963 0.00829 0.00724
## Cumulative Proportion  0.81589 0.83435 0.84640 0.85603 0.86431 0.87156
##                          PC13    PC14    PC15    PC16    PC17    PC18
## Standard deviation     0.12744 0.12543 0.11329 0.10853 0.10426 0.09972
## Proportion of Variance 0.00637 0.00617 0.00503 0.00462 0.00426 0.00390
## Cumulative Proportion  0.87793 0.88410 0.88913 0.89375 0.89801 0.90191
```

The plot of first principal component against second principal component



The plot above shows the data in a low dimension scatter plot. We can see some cluster in the middle, but they are not too apparent. Also, we got all principal components with standard deviation and variance:




After the PCA, we choose the top 2 principal components to do k-means cluster since the data only have 820 subjects. The cluster result is shown below, and the result gives the best minimum points value is 3:

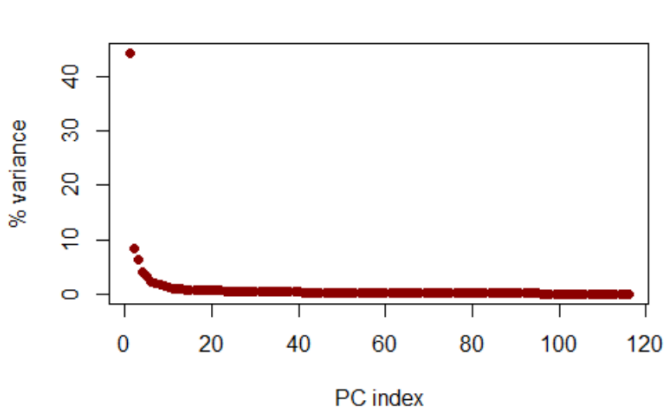
| | i | cluster | noise |
|---|---|---|---|
| 1 | 2 | 482 | 215 |
| 2 | 3 | 216 | 504 |
| 3 | 4 | 118 | 709 |
| 4 | 5 | 79 | 891 |
| 5 | 6 | 2 | 52 |
| 6 | 7 | 2 | 57 |
| 7 | 8 | 2 | 288 |

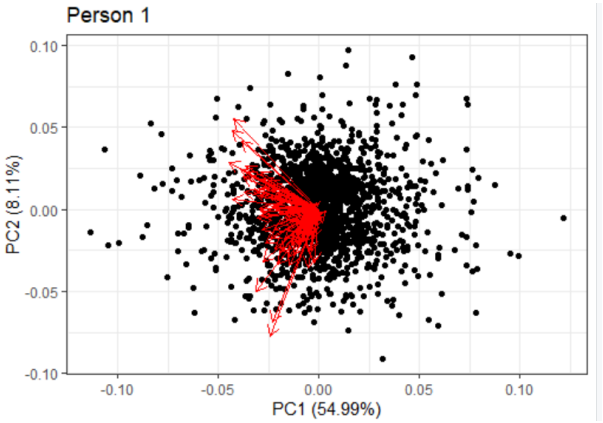
Combined the cluster result with the top 2 principal components and then group the 820 persons by cluster after deleting the noise point.

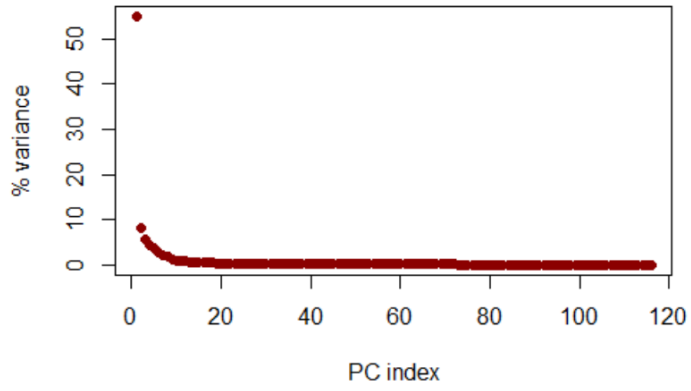
```
> final_p
# A tibble: 1,136 x 5
# Groups:   cluster [216]
       PC1     PC2 cluster pc1_mean pc2_mean
     <db l>   <db l>   <db l>    <db l>    <db l>
 1  0.123    0.420       216   0.138     0.385
 2  0.745    0.223       152   0.744     0.208
 3  0.111   -0.0507      201   0.0848   -0.0281
 4  0.420   -0.270       148   0.415    -0.318
 5  0.0858  -0.0451      201   0.0848   -0.0281
 6 -1.87    -0.267        16  -1.88     -0.283
 7 -0.188   -0.981        14  -0.254    -1.02
 8 -0.979    0.147        74  -0.961     0.0902
 9 -1.28    -0.243        64  -1.33     -0.256
10 -1.94     0.0307       20  -1.89      0.0175
# ... with 1,126 more rows
```

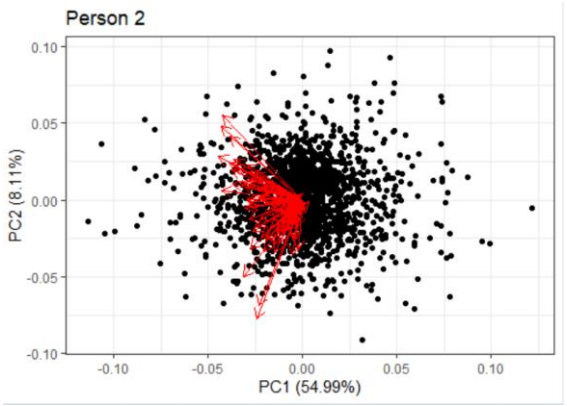
The table above shows the first two principal components with the cluster and means of each pc and the best min point value is 3.

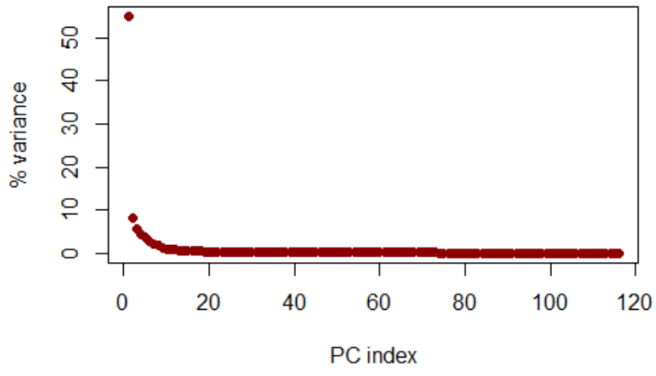
Since there are too many subjects in the data and hard to find more useful information, we need to look at the PCA of each region for each person. We did the first 3 person with their regions.

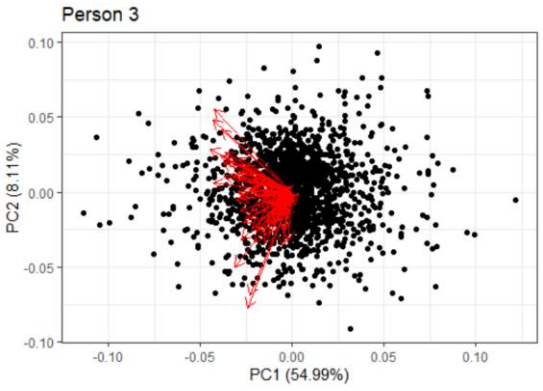


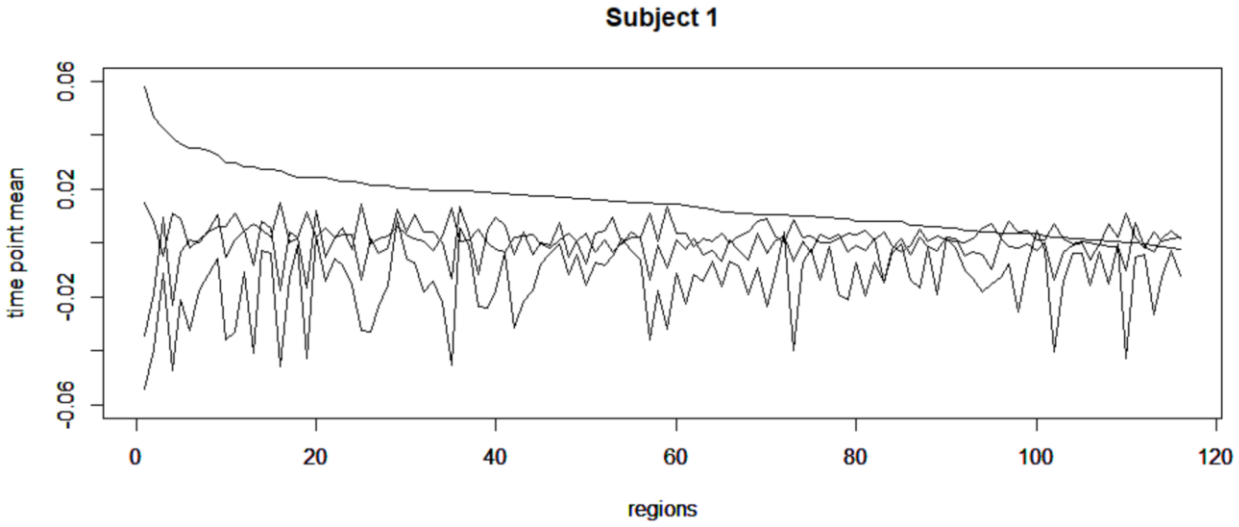
From above variance plots about the PCA for first three people with their each region, the variance is more significant in the first 20 regions since the PC index responses to regions' PCA for each person. And the scales of top 2 principal components plots for each person also show that it is more significant for first several regions.

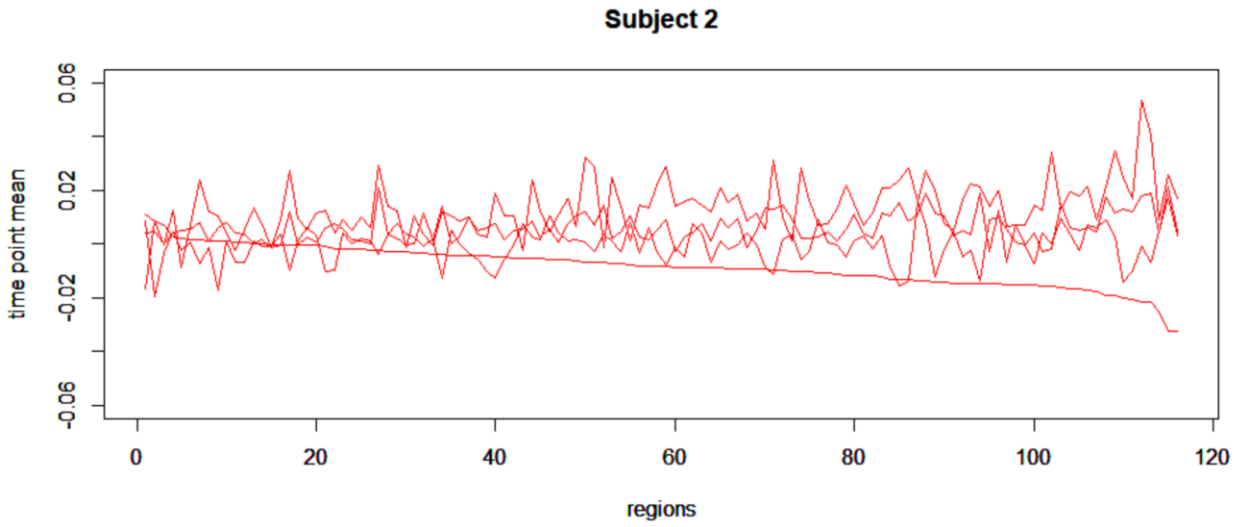
So, the next step is to find the relationship of regions in all four scans for each person.

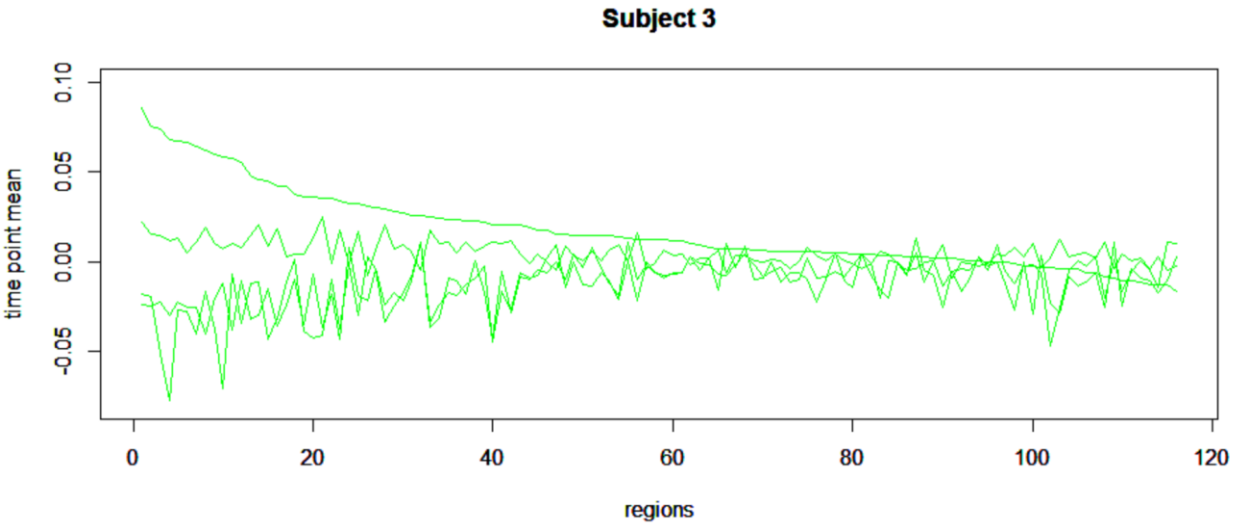
## Part 4. Analysis for 4 scans each subject with order of regions

From the Part 2 result, the last period time points are more significant, so we took last 120 time points of 1200 which is from 1081 to 1200 time points with all four scans each subject to analyze.

We did the first 3 subjects as an example to see if there is any relationship between 120 time points and 116 regions in four scans for each subject. We first took average on each scan's 120 time points and ordered the first scan's 116 regions which from largest to smallest based on the time points; and after we got the order of regions in first scan, we ordered the other three scans based on the first scan's region-order for each subject. By plotting out 120 time points against 116 ordered regions, we put 4 scans for one subject in the same plot:

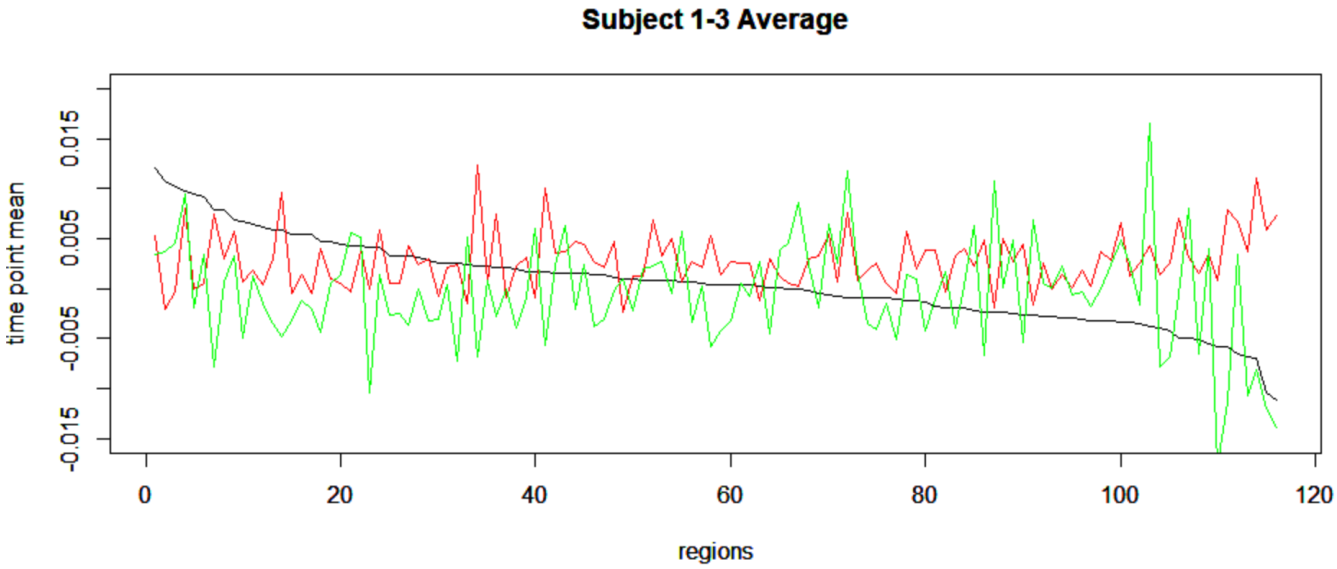





From above plots, we can find that there is a decreasing line which is for ordered regions in first scan for each subject and the other three lines are the scans based on the ordered regions in first scan. The decreasing lines in the plots are ordered from largest to smallest and the other three lines are in random shape for each subject. We may consider that there is no significant relationship between 120 time points and 116 regions for each subject and every subject are not same with other subject.

After this, we combined all four scans and took average for each subject to see the relationship between time points with regions. We used the first subject's average of four scans to order the 116 regions as a baseline and other two subjects regions' order are based on the same order of first subject. We put the average of four scans in one plot since they are in the same region order.




After taking the average of all four scans for each subject, the decreasing line is for first subject's average four scans in decreasing order of regions from largest to smallest, and the other two lines used the same region order and the result is almost the same with previous plots: there is no significant relationship between 120 time points and 116 regions for each subject and every subject are not same with other subject.

# Conclusion

In this thesis, after preliminary analyze the data we have, we first analyzed 820 subjects' first scan for 3 steps: 1. Cross validation to find the smallest error term and penalty parameter; 2. Get object value based on the LRMGLM model; 3. Check the values for function. From this part of analysis, we found that the last 120 time points (1081 time points to 1200 time points) is more significant.

The Principal Component Analysis (PCA) also gave us some information about the regions. From variance plots about the PCA for first three people with their each region, the variance is more significant in the first 20 regions. And the scales of top 2 principal components plots for each person also show that it is more significant for first several regions.

The last step is to find the relationship between 116 regions and 120 time points in all four scans for each subject. We ordered the regions and average the four scans for each subject. The result shows that the 120 time points against 116 regions do not have a significant relationship between each other in four scans, and also after taking average of four scans, there is still no relationship between time points and regions, and it shows a random order. So, we can conclude that there is no significant relationship between 120 time points and 116 regions for each subject and every subject are not same with other subject.

# Limitation and Future Work

This thesis is work on finding the relationship between time points and regions of human brain and based on the previous work from MATLAB, so the choices of method and parameters of function are according to the same way in MATLAB. Since the whole dataset are too large even for the RIT sever, we began with the first scan for total 820 subjects and the first several regions with small time period, there may be missing some important information to discover the result and relationship. Also, we did on the fMRI dataset, but not combine the Brain Subjects' information with the fMRI data.

Therefore, one of the future works is that find the relationship between more regions and more time points with all four scans for each subject. And combine the subject's information with their fMRI data to find if there are any relationships of fMRI data with the gender, age or other categories.

The main objective of this thesis is to give a new perspective on statistical research and analyze on human brain activities with regions and time points, there still have more availability, space and more effective way.

# Reference

1. Genon, S., Reid, A., Langner, R., Amunts, K., & Eickhoff, S. B. (2018, February 28). How to Characterize the Function of a Brain Region. Trends in Cognitive Sciences Volume 22, ISSUE 4, P350-364. DOI:https://doi.org/10.1016/j.tics.2018.01.010

2. How to measure brain activity in people (2018, March 29)... Retrieved from https://qbi.uq.edu.au/brain/brain-functions/how-measure-brain-activity-people.

3. Smith, K. (2012, April 4). Brain imaging: fMRI 2.0. Nature 484, 24-26. doi:10.1038/484024a

4. Devlin, H., Clare, S., & Tracey, I. Introduction to FMRI. Nuffield Department of Clinical Neurosciences, Medical Sciences Division.

5. Rogers, B. P., Morgan, V. L., Newton, A. T., & Gore, J. C. (2007). Assessing functional connectivity in the human brain by fMRI. Magnetic resonance imaging, 25(10), 1347–1357. doi:10.1016/j.mri.2007.03.007.

6. Zhang, T., Pham, M., Sun, J., Yan, G., Li, H., Sun, Y., … Coan, J. A. (2018). A low-rank multivariate general linear model for multi-subject fMRI data and a non-convex optimization algorithm for brain response comparison. NeuroImage, 173, 580–591. doi: 10.1016/j.neuroimage.2017.12.032

7. Yang, Yidan, "Low-Rank Multivariate General Linear Model and One-Way Random Effect Models for Brain Response Analysis" (2019). Thesis. Rochester Institute of Technology.

# Appendix

## R Code

```
# Preliminary Analysis
## Read fMRI data
load("Scans.arr")
## Dimensions of data
dim(Scans.arr)

## Example of 1st subject with the first 4 time poins
## and first 5 regions in all 4 scans
Scans.arr[ 1:4, 1:5, 1:4, 1]

## Example of first 3 subjects with the first 4 time points
## and first 5 regions in their first scan
Scans.arr[ 1:4, 1:5, 1, 1:3]

## Each subject's information
load("Brain.subjects.df")
list(Brain.subjects.df)

# Optimization
## Define from MATLAB code
### d is random array with dimension (7,116,820)
### u is random matrix with dimension (33,p) where p=2 in this case
### v is random array with dimension (p,116,820) where p=2
### here n = 820
### here J = dim(Y_train[,,1])[2]=116
```

```
## lr function
lr.func.stp12 <- function(Y, D, d, B, u, v, lambda, R){
  # [T,L]=size(X)
  mat.TL <- matrix(c(dim(B)[1], dim(B)[2]), nrow = 1)
  Tn = mat.TL[1]
  Ln = mat.TL[2]
  P = dim(u)[2]
  Iden_T = Matrix(diag(Tn), sparse = T)
  Iden_T <- as.matrix(Iden_T)
  H2 = inv(t(D)%*%D)%*%t(D)
  H = D%*%H2
  ITH = Iden_T - H
  totalR = kronecker(diag(P), R)
  F_cur=obj_val(Y,D,d,B,u,v,lambda,R)
  maxIter=20
  N = dim(Y)[3] # Y_train
  ########################################
  # the following for loop is for rest of function
  for (iter in 1:maxIter)
  {
    # U and d are fixed, find V
    W = B%*%u
    Y_new <- array(data = NA, dim = c(dim(Y)[1],dim(Y)[2],dim(Y)[3]))
    for (i in 1:dim(Y)[3]) {
      Y_new[,,i] = Y[,,i]-D%*%d[,,i]
    }
    Q = t(W)%*%W
```

```
inv_Q = inv(Q)
for (i in 1:dim(Y)[3]) {
  v[,,i] = inv_Q%*%(t(W)%*%Y_new[,,i])
}
#v
#}
###################################
# When V are fixed,
# min_{U} 1/n ||Y -Dd - X*U*V||^2 + lambda U' R U
# Calculate sum of V_iV_i^T and XY_iV_i^T
totalW = matrix(0, nrow = Ln*P, ncol = Ln*P)
f = matrix(0, nrow = Ln*P, ncol = 1)
for (n in 1:N) {
  temp = t(B)%*%ITH
  VVT = v[,,n]%*%t(v[,,n])
  XTX = temp%*%B
  totalW = totalW+ kronecker(VVT,XTX)
  f = f + matrix(temp%*%Y[,,n]%*%t(v[,,n]), nrow = P*Ln, ncol = 1)
}
Q = totalW/N + lambda*totalR
f=f/N
# pcg part
# [sol,flag]=pcg(Q,f,1e-5,2000)
sol <- pcg(Q, f, maxiter = 5000, tol = 1e-06)
u = matrix(sol, nrow = Ln, ncol = P)
# in Matlab d_cur=cell(N,1)
# d = array(data = sol, dim = c(7,116,820))
d = array(data = sol, dim = c(7,116,N))
```

```
    for (n in 1:N) {
      d[,,n] = H2%*%Y[,,n]
      d[,,n] = d[,,n] - H2%*%B%*%u%*%v[,,n]
    }
    F_new=obj_val(Y,D,d,B,u,v,lambda,R)
    if (F_new - F_cur > 1e-5){
      F_cur = F_new
      break}
    F_cur =  F_new
  }
  return(list(d=d,v=v,u=u))
}


## Predict Curve Function
predict_curve <- function(Y_test, D, B , U_cur){
  N <- dim(Y_test)[3]
  P1 <- ncol(U_cur)
  W <- B%*%U_cur
  A <- cbind(W,D)
  inv_A = solve(t(A)%*%A)
  temp=0
  # in Matlab d_test=cell(N,1), V_test=cell(N,1)
  d_test <- array(data = 0, dim = c(7,116,N))
  V_test <- array(data = 0, dim = c(P1,116,N))
  for (n in 1:N) {
    sol = inv_A%*%(t(A)%*%Y_test[,,n])
    V_test[,,n] = sol[1:P1,]
    d_test[,,n] = sol[-c(1:P1),]
```

```
  }
  for (i in 1:dim(Y_test)[3]) {
    temp = temp + norm(Y_test[,,i]-D%*%d_test[,,i],'F')^2
  }
  error = temp/N
  return(list(d_test=d_test,V_test=V_test,error=error))
}

## Cross Validation
library('R.matlab')
library('matrixcalc')
### Read data from basis from MATLAB
basis <- readMat("basis.mat")
### Load for 820 subjects' first scan
load("Y_oneScan")
Y <- Y_oneScan
D <- basis$D
B <- basis$B
R <- basis$R
P <- matrix(data = 4, nrow = 1)

lambda1 <- matrix(data = -1:15, nrow = 1)
lambda1 <- exp(lambda1)
error <- matrix(data = 0, nrow = length(P), ncol = length(lambda1))
for (fold in 1:5) {
  # Split data into training and testing
  index = sample(rep(1:5, 820/5))
  train = which(index!=fold)
```

```
test =  which(index==fold)
Y_train = Y[,,train]
Y_test = Y[,,test]
###
# Choose the number of principal curves
n = dim(Y_train)[3]
T = dim(Y_train[,,1])[1]
J = dim(Y_train[,,1])[2]
T = dim(B)[1]
L = dim(B)[2]
r = dim(D)[2]
# Initilize starting solution
# d is random array with dimension (7,116,820)
# u is random matrix with dimension (33,p) where p=2 in this case
# v is random array with dimension (p,116,820) where p=2
# here n = 820
# here J = dim(Y_train[,,1])[2]=116
d_cur = array(data = runif(r*J*n), dim = c(r,J,n))
for (p in 1:length(P)) {
  U_cur = matrix(data = runif(L*P[p]), nrow = L, ncol = P[p])
  V_cur = array(data = P[p]*J*n, dim = c(P[p],J,n))
  for (i in 1:length(lambda1)){
    lsq = lr.func.stp12(Y_train,D,d_cur,B,U_cur,V_cur,lambda=lambda1[i],R)
    d_cur = lsq$d;U_cur = lsq$u;V_cur = lsq$v
    pre = predict_curve(Y_test,D,B,U_cur)
    d_test=pre$d_test;V_test=pre$V_test;temp=pre$error
    error[p,i] = error[p,i]+temp
  }
```

```
  }
}
write.csv(error,file='test_error.csv',row.names = F)

## Object Value
obj_val <- function(Y,D,d, B,u,v, lambda, R){
  n = dim(Y)[3]
  val=0;
  for (i in 1:n) {
    val = val + norm(Y[,,i]-D%*%d[,,i]-B%*%u%*%v[,,i],'F')^2
  }
  rob.val = val/n +lambda*sum(diag(t(u)%*%(R%*%u)))
  return(rob.val)
}

op.val <- lr.func.stp12(Y, D, d, B, u, v, lambda = 11, R)
save(op.val, file = "op.val")

## Check values with function
basis <- readMat("basis.mat")
dim(basis$B)
temp = basis$B %*% op.val$u
temp
dim(temp)
plot(temp[, 1], xlab = "time points")
plot(temp[, 2], xlab = "time points")

temp2 = basis$B %*% op.val$u %*% op.val$v[,,1]
```

```
temp2
dim(temp2)
plot(temp2[,1], xlab = "time points")
plot(temp2[,2], xlab = "time points")
plot(temp2[,30], xlab = "time points")
plot(temp2[,50], xlab = "time points")
plot(temp2[,100], xlab = "time points")
plot(temp2[,116], xlab = "time points")

# Principal Component Analysis (PCA)
load("op.val")
op.v <- op.val$v
dim(op.v)
p_1 <- data.frame(t(op.v[1, , ]))
p_2 <- data.frame(t(op.v[2, , ]))
op.vpc <- rbind(p_1, p_2)

PC <- prcomp(~ ., center = T, data = op.vpc)
summary(PC)

## View in PC coordinates
library(ggplot2)
theme_set(theme_bw())

# we only need the first two principal components
top2pc <- data.frame(PC$x[,1:2])
ggplot(top2pc, aes(y = PC1, x = PC2)) + geom_point(col = 'tomato2')
## for all PCs standard deviation distribution in PC
```

```
plot(PC$sdev, col = 'red4', pch = 19, xlab = "PC index",
     ylab = 'SD')

## for all PCs variance distribution in PC
plot(100*PC$sdev^2/sum(PC$sdev^2), col = 'red4',
     pch = 19, xlab = "PC index", ylab = '% variance')

## do k-means cluster
require(data.table)
require(dbscan)

findk <- function(op.v){
  a <- numeric()
  b <- numeric()
  c <- numeric()
  for (i in 1:7){
    clust = hdbscan(op.v, minPts = i+1)
    a[i] <- i+1
    b[i] <- length(clust$cluster_scores)
    c[i] <- table(clust$cluster)[[1]]
  }
  result <- data.frame(a,b,c)
  names(result) <- c("i", "cluster", "noise")
  return(result)
}
findk(top2pc)
clust_1_3 <- hdbscan(top2pc, minPts = 3)
```

```
library(tidyverse)
final_dat <- function(op.v, clust){
  g_clust = clust$cluster
  #find the cluster result
  finalData = cbind(op.v, cluster=g_clust)
  #combine the cluster result with the top 2 PCs
  finalData <- finalData[-which(finalData$cluster == 0), ]
  #delete the noise point
  finalData <- finalData %>%
    group_by(cluster) %>%
    #group the 820 persons by cluster
    mutate(pc1_mean = mean(PC1), pc2_mean = mean(PC2))
  #add two new variables to the final data result,
  #which is calculate the mean of top 2 PC for each cluster
}
final_p <- final_dat(top2pc, clust_1_3)
write.table(final_p, file="final_p.txt", sep="\t", quote=F, row.names=F)

## PCA for each person with each region
library(ggfortify)
p1PC <- prcomp(~ ., center = T, data = op.vpc[1&821, 1:116])
summary(p1PC)
plot(100*p1PC$sdev^2/sum(p1PC$sdev^2), col = 'red4',
     pch = 19, xlab = "PC index", ylab = '% variance')
autoplot(p1PC, loadings = TRUE, main = "Person 1")
p2PC <- prcomp(~ ., center = T, data = op.vpc[2&822, 1:116])
summary(p2PC)
plot(100*p2PC$sdev^2/sum(p2PC$sdev^2), col = 'red4',
```

```
     pch = 19, xlab = "PC index", ylab = '% variance')
autoplot(p2PC, loadings = TRUE, main = "Person 2")
p3PC <- prcomp(~ ., center = T, data = op.vpc[3&823, 1:116])
summary(p3PC)
plot(100*p3PC$sdev^2/sum(p3PC$sdev^2), col = 'red4',
     pch = 19, xlab = "PC index", ylab = '% variance')
autoplot(p3PC, loadings = TRUE, main = "Person 3")

# Analysis for 4 scans each subject with order of regions
load("Scans.arr")
scan <- Scans.arr

# Person 1
## Person 1 Scan 1
p1_s1 <- scan[ 1081:1200, 1:116, 1, 1]

rnames = paste("t", 1:120, sep = "")
rownames(p1_s1) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p1_s1) = cnames

p1s1 <- t(as.matrix(colSums(p1_s1)/120))
p1s1o <- p1s1[,order(p1s1[1,], decreasing = T)]
plot(p1s1o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "black", main = "Subject 1", ylim = c(-0.06, 0.06))



## Person 1 Scan 2
```

```
par(new = TRUE)
p1_s2 <- scan[ 1081:1200, 1:116, 2, 1]
rnames = paste("t", 1:120, sep = "")
rownames(p1_s2) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p1_s2) = cnames

p1s2 <- t(as.matrix(colSums(p1_s2)/120))
p1s2o <- p1s2[,order(p1s1[1,], decreasing = T)]
plot(p1s2o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "black", ylim = c(-0.06, 0.06))

## Person 1 Scan 3
par(new = TRUE)
p1_s3 <- scan[ 1081:1200, 1:116, 3, 1]
rnames = paste("t", 1:120, sep = "")
rownames(p1_s3) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p1_s3) = cnames

p1s3 <- t(as.matrix(colSums(p1_s3)/120))
p1s3o <- p1s3[,order(p1s1[1,], decreasing = T)]
plot(p1s3o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "black", ylim = c(-0.06, 0.06))

## Person 1 Scan 4
par(new = TRUE)
p1_s4 <- scan[ 1081:1200, 1:116, 4, 1]
```

```
rnames = paste("t", 1:120, sep = "")
rownames(p1_s4) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p1_s4) = cnames

p1s4 <- t(as.matrix(colSums(p1_s4)/120))
p1s4o <- p1s4[,order(p1s1[1,], decreasing = T)]
plot(p1s4o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "black", ylim = c(-0.06, 0.06))



# Person 2
## Person 2 Scan 1
p2_s1 <- scan[ 1081:1200, 1:116, 1, 2]

rnames = paste("t", 1:120, sep = "")
rownames(p2_s1) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p2_s1) = cnames

p2s1 <- t(as.matrix(colSums(p2_s1)/120))
p2s1o <- p2s1[,order(p2s1[1,], decreasing = T)]
plot(p2s1o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "red", main = "Subject 2", ylim = c(-0.06, 0.06))



## Person 2 Scan 2
par(new = TRUE)
```

```
p2_s2 <- scan[ 1081:1200, 1:116, 2, 2]


rnames = paste("t", 1:120, sep = "")
rownames(p2_s2) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p2_s2) = cnames

p2s2 <- t(as.matrix(colSums(p2_s2)/120))
p2s2o <- p2s2[,order(p2s1[1,], decreasing = T)]
plot(p2s2o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "red", main = "Subject 2", ylim = c(-0.06, 0.06))

## Person 2 Scan 3
par(new = TRUE)
p2_s3 <- scan[ 1081:1200, 1:116, 3, 2]
rnames = paste("t", 1:120, sep = "")
rownames(p2_s3) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p2_s3) = cnames

p2s3 <- t(as.matrix(colSums(p2_s3)/120))
p2s3o <- p2s3[,order(p2s1[1,], decreasing = T)]
plot(p2s3o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "red", ylim = c(-0.06, 0.06))

## Person 2 Scan 4
par(new = TRUE)
p2_s4 <- scan[ 1081:1200, 1:116, 4, 2]
```

```
rnames = paste("t", 1:120, sep = "")
rownames(p2_s4) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p2_s4) = cnames

p2s4 <- t(as.matrix(colSums(p2_s4)/120))
p2s4o <- p2s4[,order(p2s1[1,], decreasing = T)]
plot(p2s4o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "red", ylim = c(-0.06, 0.06))



# Person 3
## Person 3 Scan 1
p3_s1 <- scan[ 1081:1200, 1:116, 1, 3]

rnames = paste("t", 1:120, sep = "")
rownames(p3_s1) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p3_s1) = cnames

p3s1 <- t(as.matrix(colSums(p3_s1)/120))
p3s1o <- p3s1[,order(p3s1[1,], decreasing = T)]
plot(p3s1o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "green", main = "Subject 3", ylim = c(-0.08, 0.1))



## Person 3 Scan 2
par(new = TRUE)
```

```
p3_s2 <- scan[ 1081:1200, 1:116, 2, 3]


rnames = paste("t", 1:120, sep = "")
rownames(p3_s2) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p3_s2) = cnames

p3s2 <- t(as.matrix(colSums(p3_s2)/120))
p3s2o <- p3s2[,order(p3s1[1,], decreasing = T)]
plot(p3s2o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "green", main = "Subject 3", ylim = c(-0.08, 0.1))



## Person 3 Scan 3
par(new = TRUE)
p3_s3 <- scan[ 1081:1200, 1:116, 3, 3]
rnames = paste("t", 1:120, sep = "")
rownames(p3_s3) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p3_s3) = cnames

p3s3 <- t(as.matrix(colSums(p3_s3)/120))
p3s3o <- p3s3[,order(p3s1[1,], decreasing = T)]
plot(p3s3o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "green", main = "Subject 3", ylim = c(-0.08, 0.1))

## Person 3 Scan 4
par(new = TRUE)
```

```
p3_s4 <- scan[ 1081:1200, 1:116, 4, 3]
rnames = paste("t", 1:120, sep = "")
rownames(p3_s4) = rnames
cnames = paste("r", 1:116, sep ="")
colnames(p3_s4) = cnames

p3s4 <- t(as.matrix(colSums(p3_s4)/120))
p3s4o <- p3s4[,order(p3s1[1,], decreasing = T)]
plot(p3s4o, type = "l", xlab = "regions", ylab = "time point mean",
     col = "green", main = "Subject 3", ylim = c(-0.08, 0.1))

## Person 1-3 Average
mean_p1 <- (p1s1 + p1s2 + p1s3 + p1s4)/4
mean_p1o <- mean_p1[, order(mean_p1[1,], decreasing = TRUE)]
plot(mean_p1o, type = "l", col = "black", xlab = "regions",
     ylab = "time point mean", main = "Subject 1-3 Average",
     ylim = c(-0.015, 0.02))

par(new = TRUE)
mean_p2 <- (p2s1 + p2s2 + p2s3 + p2s4)/4
mean_p2o <- mean_p2[,order(mean_p1[1, ], decreasing = TRUE)]
plot(mean_p2o, type = "l", col = "red", xlab = "regions",
     ylab = "time point mean", main = "Subject 1-3 Average",
     ylim = c(-0.015, 0.02))

par(new = TRUE)
mean_p3 <- (p3s1 + p3s2 + p3s3 + p3s4)/4
mean_p3o <- mean_p3[order(mean_p1[1, ], decreasing = TRUE)]
```

```
plot(mean_p3o, type = "l", col = "green", xlab = "regions",
     ylab = "time point mean", main = "Subject 1-3 Average",
     ylim = c(-0.015, 0.02))
```